AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method comprising:

receiving a network packet;

dividing the network packet into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer;

receiving a portion of a the network packet, wherein the portion of the network packet comprises one of the plurality of m-packets;

retrieving a sequence number;

passing the sequence number to a sequence election unit;

waiting to receive a signal to process the packet from the sequence election unit; and

performing processing on the packet in response to receipt of the signal.

2. (Original) A method according to Claim 1, further comprising:

indicating to the sequence election unit that the processing is complete.

3. (Original) A method according to Claim 1, wherein the processing must be performed in group order.

4. (Original) A method according to Claim 1, wherein the processing comprises a critical section of a packet processing pipeline.

5. (Original) A method according to Claim 1, further comprising:

requesting the signal from the sequence election unit.

6. (Original) A method according to Claim 1, wherein passing the sequence number further comprises:

passing a mask associated with the sequence number to the sequence election unit.

7. (Original) A method according to Claim 6, wherein the mask indicates a group of threads designated to perform the processing.

8. (Original) A method according to Claim 7, wherein the group of threads comprises execution threads of two or more packet processing engines.

9. (Currently Amended) A medium storing program code, the program code executable to:

receive a network packet;

divide the network packet into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer;

receive a portion of ~~a~~ the network packet, wherein the portion of the network packet comprises one of the plurality of m-packets;

retrieve a sequence number;

pass the sequence number to a sequence election unit;

wait to receive a signal to process the packet from the sequence election unit; and

perform processing on the packet in response to receipt of the signal.

10. (Original) A medium according to Claim 9, wherein the program code executable to pass the sequence number comprises program code executable to:

pass a thread mask associated with the sequence number to the sequence election unit.

11. (Original) A medium according to Claim 10, wherein the thread mask indicates a group of threads designated to perform the processing.

12. (Original) A medium according to Claim 11, wherein the group of threads comprises execution threads of two or more packet processing engines.

13. (Currently Amended) A method comprising:

receiving a sequence number associated with an execution thread, wherein the sequence number is further associated with one of a plurality of m-packets, wherein an m-packet comprises a network packet divided into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer;

receiving a request for election from the execution thread;

determining whether the sequence number is less than sequence numbers associated with a respective other one or more execution threads; and

transmitting an election signal to the execution thread if it is determined that the sequence number is less than the sequence numbers associated with the respective other one or more execution threads.

14. (Original) A method according to Claim 13, further comprising:

associating the sequence number with a mask.

15. (Original) A method according to Claim 14, wherein the one or more other threads are associated with the mask.

16. (Original) A method according to Claim 15, wherein the mask indicates a group of threads designated to perform particular processing.

17. (Original) A method according to Claim 13, further comprising:

receiving an indication of completed processing from the execution thread.

18. (Original) A method according to Claim 17, further comprising:

determining a second execution thread of the other one or more execution threads that is associated with a second sequence number that is less than sequence numbers associated with each other of the other one or more execution threads; and

transmitting an election signal to the second execution thread.

19. (Original) A method according to Claim 18, wherein the one or more other threads are associated with a mask.

20. (Original) A method according to Claim 13, further comprising:

swapping out the thread if it is determined that the sequence number is not less than the sequence numbers associated with the respective other one or more execution threads.

21. (Currently Amended) A medium storing program code, the program code executable to:

receive a sequence number associated with an execution thread, wherein the sequence number is further associated with one of a plurality of m-packets, wherein an m-packet comprises a network packet divided into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer;

receive a request for election from the execution thread;

determine whether the sequence number is less than sequence numbers associated with a respective other one or more execution threads; and

transmit an election signal to the execution thread if it is determined that the sequence number is less than the sequence numbers associated with the respective other one or more execution threads.

22. (Original) A medium according to Claim 21, the program code further executable to:

receive an indication of completed processing from the execution thread;

determine a second execution thread of the other one or more execution threads that is associated with a second sequence number that is less than sequence numbers associated with each other of the other one or more execution threads; and

transmit an election signal to the second execution thread.

23. (Original) A medium according to Claim 21, the program code further executable to:

swap out the thread if it is determined that the sequence number is not less than the sequence numbers associated with the respective other one or more execution threads.

24. (Currently Amended) A method comprising:

receiving a sequence number associated with an execution thread, wherein the sequence number is further associated with one of a plurality of m-packets, wherein an m-packet comprises a network packet divided into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer;

determining whether a processing sequence associated with the sequence number is locked; and

transmitting an election signal to the execution thread if it is determined that the processing sequence is not locked.

25. (Original) A method according to Claim 24, further comprising:

receiving an indication of completed processing from the execution thread.

26. (Original) A method according to Claim 25, further comprising:

determining a second execution thread of another one or more execution threads that is associated with a second sequence number that is less than sequence numbers associated with each other of the other one or more execution threads; and

transmitting an election signal to the second execution thread.

27. (Original) A method according to Claim 26, wherein the execution thread and the another one or more threads are associated with a mask.

28. (Original) A method according to Claim 26, wherein the another one or more threads are designated as stalled threads from which respective sequence numbers were previously received.

29. (Original) A method according to Claim 24, further comprising:

designating the execution thread as a stalled thread if it is determined that the processing sequence is locked.

30. (Original) A method according to Claim 24, wherein the processing sequence comprises a critical section of a packet processing pipeline.

31. (Currently Amended) A medium storing program code, the program code executable to:

receive a sequence number associated with an execution thread, wherein the sequence number is further associated with one of a plurality of m-packets, wherein an m-packet

comprises a network packet divided into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer;

determine whether a processing sequence associated with the sequence number is locked; and

transmit an election signal to the execution thread if it is determined that the processing sequence is not locked.

32. (Original) A medium according to Claim 31, the program code further executable to:

receive an indication of completed processing from the execution thread;

determine a second execution thread of another one or more execution threads that is associated with a second sequence number that is less than sequence numbers associated with each other of the other one or more execution threads; and

transmit an election signal to the second execution thread.

33. (Currently Amended) A device comprising:

a packet processing engine to provide a plurality of execution threads;

a sequence number register to provide a respective sequence number to one or more of the plurality of execution threads, wherein the respective sequence number is further associated with one of a plurality of m-packets, wherein an m-packet comprises a network packet divided into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer; and

a sequence election unit to elect one of the one or more of the plurality of execution threads based on sequence numbers provided to the one or more of the plurality of execution threads.

34. (Original) A device according to Claim 33, the sequence election unit further to transmit an election signal to the elected one of the one or more of the plurality of execution threads.

35. (Original) A device according to Claim 33, the sequence election unit further to associate the one or more of the plurality of execution threads with a mask.

36. (Original) A device according to Claim 35, wherein the mask indicates a group of threads designated to perform particular processing.

37. (Original) A device according to Claim 33, the sequence election unit further to receive a sequence number associated with an execution thread of the one or more of the plurality of execution threads, to determine whether a processing sequence associated with the sequence number is locked, and to transmit an election signal to the execution thread if it is determined that the processing sequence is not locked.

38. (Original) A device according to Claim 37, the sequence election unit further to receive an indication of completed processing from the execution thread, to determine a second execution thread of the one or more of the plurality of execution threads that is associated with a second sequence number that is less than sequence numbers associated with each other of the

one or more of the plurality of execution threads, and to transmit an election signal to the second execution thread.

39. (Original) A device according to Claim 33, further comprising a network processor comprising the packet processing engine, the sequence number register, and the sequence election unit.

40. (Original) A system according to Claim 39, wherein the network processor further comprises two or more packet processing engines.

41. (Currently Amended) A system comprising:

a packet processing engine to provide a plurality of execution threads;

a sequence number register to provide a respective sequence number to one or more of the plurality of execution threads, wherein the sequence number is further associated with one of a plurality of m-packets, wherein an m-packet comprises a network packet divided into a plurality of m-packets, wherein a size of each of the plurality of m-packets is equal to an element size, and wherein the element size is associated with a receive buffer;

a sequence election unit to elect one of the one or more of the plurality of execution threads based on sequence numbers provided to the one or more of the plurality of execution threads; and

a Double Data Rate memory.

42. (Original) A system according to Claim 41, further comprising a network processor comprising the packet processing engine, the sequence number register, and the sequence election unit.

43. (Original) A system according to Claim 42, wherein the network processor further comprises two or more packet processing engines.

44. (Original) A system according to Claim 41, the sequence election unit further to receive a sequence number associated with an execution thread of the one or more of the plurality of execution threads, to determine whether a processing sequence associated with the sequence number is locked, to transmit an election signal to the execution thread if it is determined that the processing sequence is not locked, to receive an indication of completed processing from the execution thread, to determine a second execution thread of the one or more of the plurality of execution threads that is associated with a second sequence number that is less than sequence numbers associated with each other of the one or more of the plurality of execution threads, and to transmit an election signal to the second execution thread.